I/S: A JOURNAL OF LAW AND POLICY FOR THE INFORMATION SOCIETY

MAX HENRION*

Open-Source Policy Modeling

Abstract: Computer models are widely used by governments to analyze, predict, and evaluate the benefits, risks, and costs of public policy and regulations. This paper explores the proposal that all computer models used by governments to evaluate or justify public policy should be *open source*; that is, the source program code in which the models are written should be publicly available for everyone to download, review, run, and modify. This proposal is inspired by the remarkable success of open-source products, such as Linux, Apache, MySQL, and Firefox, and open content projects, such as Wikipedia and the open directory project. Open-source policy modeling has important similarities and differences with existing open-source projects. It could significantly improve the transparency, reliability, and reusability of policy models. It could also lead to a more collaborative development process with a wider range of contributors. Arguably, existing guidelines for regulatory analysis from the White House's Office of Management and Budget already imply something like this, subject to limitations for proprietary software, confidentiality, and security. There will be benefits and challenges for each kind of organization affected: government agencies, stakeholder groups including industry associations, environmental and social justice groups, community groups, universities and nonprofit think-tanks, consulting analysts and firms, and the general citizen. Over time, the ripple effects of open source modeling have the potential to profoundly transform the way we make public policy.

* The author is employed at Lumina Decision Systems, Inc. and can be reached at henrion@lumina.com.

I. COMPUTER MODELS FOR POLICY ANALYSIS

Governments use quantitative computer models ever more widely to evaluate and justify current and proposed policies. Areas in which public agencies use models include environmental regulation, education, transportation, justice, defense, budgets, and taxes. In some cases, model-based analysis is mandatory. For example, in the United States, the Office of Management and Budget (OMB) requires federal regulatory agencies to perform a quantitative cost-benefit analysis of any proposed new or changed rule that is "economically significant."[1]

Some policy models are constructed and executed by government analysts, and others by an expanding ecosystem of consultants who conduct analyses for their government clients. Policy models may be spreadsheets, either small, large, or massive. They may be built in visual simulation tools, such as Analytica[2] or Stella,[3] or statistical packages, such as SAS.[4] The most complex models are typically custom-built computer programs built in a conventional computer language, such as Fortran or C++.

Ideally, these models provide policymakers with a deeper understanding of, and more accurate estimates concerning, the effects of proposed policies; they lead to better decisions than the more traditional seat-of-the-pants approach to decision making. But, model-based policymaking also brings risks. Models may contain unrecognized errors, such as programming bugs, omissions, and biases, either deliberate or unconscious, which arise from input assumptions, model formulation, and the analysts' perspectives. The technical complexities of computer models can impede communication between policymakers and those stakeholders who lack the expertise needed to understand model assumptions and implications. This gulf increases the danger that public policy will be disproportionately influenced by lobbyists who have the resources to hire the appropriate specialists. Due to the technical complexity of the subject matter, stakeholders without such resources may be disenfranchised.

---

[1] As defined in Executive Order 12866 §3(f)(1).

[2] Lumina Decision Systems, Inc., http://www.lumina.com (accessed November 7, 2007).

[3] ISEE Systems, Inc., http://www.iseesystems.com (accessed November 7, 2007).

[4] SAS Institute, Inc., http://www.sas.com (accessed November 7, 2007).

II. MODEST PROPOSAL

My purpose here is to introduce a modest proposal intended to reduce these risks:

> All computer models used by government to evaluate or justify public policy should be open source: That is, the source program code in which the models are written should be publicly available for anyone to download, review, run, and modify.

This proposal is applicable to policy models used by any agency at any level of government — local, state, regional, national — as well as international organizations such as the European Union and the United Nations. It is relevant to all forms of public policy, including legislation, regulations, taxes, and budgets.

It does *not* require that policy models use open-source modeling software. It is neither practical nor necessary to require that modelers abandon proprietary applications such as Microsoft Excel, Analytica,[5] or, in the case of custom software, computer languages with proprietary compilers and interactive development environments. The important thing is that the source code of the model — for example, the spreadsheet formulas, model equations, and procedural code — is accessible, and that the proprietary software used to run it is readily available and not prohibitively expensive.

III. OPEN-SOURCE SOFTWARE

This proposal is inspired by the open-source software movement and related open community content projects, such as Wikipedia,[6] the online encyclopedia. Open-source policy models are implemented as computer programs so they are a kind of software, but with important differences from other kinds of software.

Source code refers to the set of instructions written by a programmer or modeler using a computer language, a visual modeling tool, or spreadsheet formulas, that create a program or model. *Open-source software* is software with licensing provisions that permit

---

[5] As the originator of the proprietary modeling software, Analytica (Lumina, 2005), I acknowledge a special interest here.

[6] Wikipedia, http://www.wikipedia.org (accessed November 7, 2007).

anyone to download, review, modify, improve, and redistribute the source code. In contrast, the writers and publishers of conventional proprietary software often go to great pains to make sure that no one else can access their source code. There are a variety of open-source licenses that specify how source code may be redistributed and how contributors are acknowledged.

Open-source software products have been remarkably successful; most of the software managing the Internet's infrastructure is open source. Exact numbers are unreliable, but according to widely published estimates, Apache has approximately 50% of the global web server market.[7] MySQL accounts for about 29% of database systems, and the Firefox web browser encompasses about 25% of global market share.[8]

Open-source software is not necessarily free of charge. Several companies, such as Red Hat Software and MySQL, have grown successful businesses selling open-source software along with documentation, support, and certifications, and have shown that customers are willing to pay for this additional value. The ability of others to redistribute open-source software limits the price the companies can charge for the software without the extra services.

IV. OMB GUIDELINES FOR TRANSPARENCY AND REPRODUCIBILITY

In recent years, the United States Office of Information and Regulatory Affairs (OIRA) of the OMB has issued a series of directives aimed at improving the quality of information and analysis used by the government as a basis for its public policy. The *Data Quality Guidelines* issued by the OMB establish standards of transparency and reproducibility for the information and analysis used as a basis for regulations.[9] The OMB's *Circular A-4* specifies methods for "good" regulatory analysis that include quantification of

---

[7] "July 2007 Web Server Survey," *Netcraft LTD*, Novermber 1, 2007, http://news.netcraft.com/archives/2007/07/09/july_2007_web_server_survey.html (accessed November 7, 2007).

[8] Percy Cabello, "W3Counter: Firefox Holds 25% of Browser Market," *Mozilla Links*, June 29, 2007, http://mozillalinks.org/wp/2007/06/w3counter-firefox-holds-25-of-browser-market/ (accessed November 7, 2007).

[9] Office of Management and Budget, "Guidelines for Ensuring and Maximizing the Quality, Objectivity, Utility, and Integrity of Information Disseminated by Federal Agencies," Federal Register 67, no. 36 (February 2002), http://www.whitehouse.gov/omb/fedreg/reproducible2.pdf.

costs and benefits, probabilistic treatment of uncertainties, and sensitivity analysis.[10] The OMB extends the data quality guidelines with requirements for the conduct of peer review.[11] Most of these were developed under the leadership of John Graham, the Administrator of the OMB from 2001 to 2006.

The recent *Proposed Risk Assessment Bulletin* gives technical guidance for quantitative risk assessments by the federal government:

> A risk assessment report should also have a high degree of transparency with respect to data, assumptions, and methods that have been considered. Transparency will increase the credibility of the risk assessment, and will allow interested individuals, internal and external to the agency, to understand better the technical basis of the assessment.
>
> Influential[12] risk assessments should be capable of being substantially reproduced. . . . [T]his means that independent reanalysis of the original or supporting data using the same methods would generate similar analytical results, subject to

---

[10] Office of Management and Budget, *Circular A-4: Regulatory Analysis*, "Transparency and Reproducibility of Results," September 17, 2003 (Washington, D.C.), § E (4), http://www.whitehouse.gov/omb/inforeg/circular_a4.pdf (guidance to assist analysts in the regulatory agencies by defining good regulatory analysis); Office of Management and Budget, *Final Information Quality Bulletin for Peer Review*, December 15, 2004 (Washington, D.C.), http://www.whitehouse.gov/omb/inforeg/peer2004/peer_bulletin.pdf. I cannot but welcome most of these guidelines, having long advocated that policy analysts should borrow more extensively from the standard practices of science. Max Henrion, "Computer Aids For a Dialectical Approach to Designing Policy Models," in *Design Policy: Design and Information Technology,* ed. Richard Langdon and George Mallen (Design Council, 1985): 53–61. Granger Morgan and I had the temerity to propose "ten commandments for good policy analysis" with which the OMB guidelines have a gratifying overlap. M. Granger Morgan and Max Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis* (New York: Cambridge University Press, 1990): 36–43.

[11] OMB, *Circular A-4*, 17.

[12] OMB defines an assessment as *influential*, and so subject to this requirement, if it has potential impact of more than $500 million in any year, or "is novel, controversial, precedent-setting, or has significant interagency interest." OMB, *Circular A-4*, 22.

> an acceptable degree of precision. Public access to original data is necessary to satisfy this standard . . . .[13]

The OMB guidelines do not explicitly address the issue of model source code, but as a practical matter, it is hard to see how an analysis using a computer model could be transparent and reproducible without releasing its source code. In principle, model authors could publish model specifications sufficiently explicit to enable a reviewer to reconstruct the model, but that would be a lot of extra work for the model authors and even more work for reviewers who would need to recreate the model. In any case, without the original source code, it would be impossible for reviewers to determine if any discrepancy was due to inaccurate specifications, errors in the original implementation, or errors in the reproduction.

The OMB recognizes that some models may be proprietary or contain confidential or proprietary information, and admits those as compelling reasons not to publish models.[14] In the US, models developed using public funds are in the public domain, and may, in principle, be obtained via the Freedom of Information Act (FOIA), subject to similar exclusions and, of course, exigencies of national security. Subject to these restrictions, the OMB guidelines come close to implying open-source policy models but balk at the final jump.

V. DETECTING AND CORRECTING ERRORS

Errors may be more prevalent in policy models than is generally recognized. Naturally, few model authors wish to publicize the errors they find. The European Spreadsheet Risks Interest Group (EUSPRIG) has compiled a list of news stories reporting disasters due to errors in spreadsheets.[15] A number of empirical studies based on audits of operational spreadsheets in government and business find that up to 90% of spreadsheets contain serious errors with respect to the intentions of their authors.[16] Spreadsheet authors and users are

---

[13] Office of Management and Budget, *Proposed Risk Assessment Bulletin*, 14–17, January 9 2006, http://www.whitehouse.gov/omb/inforeg/proposed_risk_assessment_bulletin_010906.pdf.

[14] This point is discussed further in section VIII of this article.

[15] European Spreadsheet Risks Interest Group, "Spreadsheet Mistakes – News Stories," *EuSpRIG.org*, http://www.eusprig.org/stories.htm (accessed November 7, 2007).

typically unaware of these findings and are unjustifiably confident about the reliability of their models.

Part of the problem is that many spreadsheet errors are easy to make and hard to detect. Formulas using meaningless cell references, such as A1 or BX103, are much harder to understand and verify than expressions using meaningful variable names. Often there is no way to check the results produced by the model against real-world results because the values being modeled are forecasts or are otherwise not directly observable. If an error leads to results that are off by 25%, it may not be obvious, unlike bugs in conventional software that create obviously wrong behavior or crash the system running the software. The most common way to verify models is to audit the formulas carefully and systematically. However, auditing is difficult, time consuming, unreliable at detecting all errors, and rarely performed.

Visual modeling packages detect some types of errors more effectively than spreadsheets. Visual depiction, through dependency graphs or influence diagrams, makes missing or inappropriate dependencies obvious. Array abstraction, which uses a single formula to express a mathematical relationship between array-valued variables, instead of a separate formula for each cell as in spreadsheets, can massively reduce the number of formulas to be written and checked. Even though error rates can be substantially reduced by visual modeling and array abstraction, despite the best efforts of modelers and reviewers, it is likely that all kinds of public policy models contain errors far more often than we would wish.

It is interesting and unexpected to many, including experienced software engineers, that open-source software is often more reliable and has fewer bugs and security holes than comparable proprietary software. A key reason for this seems to be the number of reviewers and developers involved in open-source software. There are more people with more perspectives looking for bugs and vulnerabilities and more people available to fix them. Famously, at least among software developers, Eric Steven Raymond wrote "[g]iven enough eyeballs, all bugs are shallow," which he dubbed *Linus's Law* after Linus Torvalds, the originator of Linux.[17]

---

[16] Ramond R. Panko and Ralph H. Sprague Jr., "Hitting the Wall: Errors in Developing and Code Inspecting a 'Simple' Spreadsheet Model," *Decision Support Systems* 22, no. 4 (1998): 337–53. For a review, see Raymond R. Panko, "What We Know About Spreadsheet Errors," *Journal of End-User Computing* 10, no. 2. (1998): 15–21.

[17] Eric S. Raymond, "The Cathedral and the Bazaar," September 11, 2000, http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html (accessed November 7, 2007).

Open-source software code tends to be cleaner and clearer than closed-source software, and therefore easier to verify, maintain, and extend. This may be both because open-source programmers expect their code to be reviewed by their peers and because an incomprehensible or poorly documented code is more likely to be cleaned up or replaced by someone else. We can reasonably hope for similar benefits — cleaner code and faster detection and fixing of errors — for open-source policy models.

VI. TRANSPARENCY AND SENSITIVITY ANALYSIS

Any model is necessarily a simplification of reality, so reviewers can almost always point to simplifications and omissions. Two important questions for reviewers are whether the assumptions, and plausible changes to them, will materially affect the model's conclusions and how sensitive the model is to changes in those assumptions. In its replies to comments on the Data Quality Guidelines, the OMB states:

> The primary benefit of public transparency is not necessarily that errors in analytic results will be detected, although error correction is clearly valuable. The more important benefit of transparency is that the public will be able to assess how much an agency's analytic result hinges on the specific analytic choices made by the agency. Concreteness about analytic choices allows, for example, the implications of alternative technical choices to be readily assessed. This type of sensitivity analysis is widely regarded as an essential feature of high-quality analysis, yet sensitivity analysis cannot be undertaken by outside parties unless a high degree of transparency is achieved.[18]

If reviewers can perform sensitivity analysis themselves, they may be able to shorten their list of criticisms to focus on those that could be material. This could accelerate the process by focusing discussion on matters of possible importance, thereby reducing the tendency of the review process to become bogged down on matters that turn out to be of marginal relevance.

---

[18] OMB, "Guidelines for Ensuring," 8456.

VII. IMPROVING TRANSPARENCY

Open-source does not guarantee transparency for conventional software code or for policy models. If the code is poorly documented, uses incomprehensible variable names, or ill-structured "spaghetti code," it may be difficult or impossible to understand. Writing comprehensible code takes considerable effort and skill. Ideally, all policy modelers would do this anyway to enable easier verification, maintenance, and extension of models. When a public agency engages a consultant to build a policy model, one would hope the agency would insist on clear documentation. Sadly, due to the exigencies of short deadlines, changing model objectives, and inexperienced modelers, this is not always the case.

Spreadsheet experts have recommended guidelines for improving the transparency of spreadsheets.[19] These recommendations include: sequencing the calculations like writing, from left to right and top to bottom; separating inputs, outputs, and internal computations; creating consistent documentation; using meaningful names instead of cell references; and avoiding unduly complex formulas. Unfortunately, most policy analysts pay no more attention to these recommendations than do other kinds of spreadsheet modelers.

Spreadsheets are ill-suited for creating larger models because of their limited support for modularity and for managing arrays, especially ones with more than two dimensions. It is difficult, when using a spreadsheet, to extend the time horizon, add a scenario, or modify dimensions. These characteristics inhibit reusability and extensibility, which are important factors in an evolving a family of models and a community of collaborating modelers.

Visual modeling tools, such as Analytica and Stella, use influence diagrams or systems diagrams to depict variables and their relationships as nodes and arrows. These diagrams improve transparency by offering a higher-level representation similar to an expert modeler's mental models of the problem. Unlike the flow charts in conventional software documentation, which must be manually updated to remain consistent with the source code, these diagrams constitute "live documentation" that is guaranteed to be consistent with the underlying mathematical relationships. Perhaps

---

[19] Stephen G. Powell and Kenneth R. Baker, *The Art of Modeling with Spreadsheets: Management Science, Spreadsheet Engineering, and Modeling Craft* (New Jersey: John Wiley & Sons, 2004): 91–110; John F. Raffensperger, "The New Guidelines for Writing Spreadsheets," August 20, 2003, http://www.mang.canterbury.ac.nz/people/jfraffen/spreadsheets/index.html (accessed November 7, 2007).

more important, modelers use these diagrams to design and implement the model, encouraging clear thinking and communication about model structure from the start, unlike conventional documentation, which is often written after-the-fact by someone other than the lead model designer.

There is still considerable room for improvement in the areas of transparency and documentation clarity for models of all kinds. Indeed, there is a lack of general guidelines and positive examples of transparency. A commitment to open-source policy modeling would be a substantial move toward the goal of regulatory transparency. Modelers who know that their work will be exposed to public scrutiny, especially from their peers, will have a stronger incentive to make clearly structured and documented models. If modelers fail in an open-source environment other modelers can clean up erroneous models, document them or simply replace them. Model transparency will improve over time.

VIII. CONFIDENTIALITY AND INTELLECTUAL PROPERTY

The OMB recognizes a major limitation to transparency and reproducibility:

> Public access to original data is necessary to satisfy this standard, though such access should respect confidentiality and other compelling considerations.[20]

In many cases, it is possible to preserve anonymity without losing transparency by aggregating detailed confidential data. As the OMB suggests, if it is important to review material containing unaggregated confidential data in order to review the aggregation methods for example, then it is often possible for reviewers to work under a nondisclosure agreement. Clearly, nondisclosure prevents general transparency of the open source for that element of the model.

Another "compelling consideration" is the intellectual property of model authors. It is not uncommon for regulatory agencies to use results from proprietary models created by consultants. In such cases, model owners are understandably reluctant to release the model source code. Sometimes model owners are willing to allow a restricted release under a nondisclosure agreement to reviewers approved by the model owners as noncompetitive. This example of the status quo is

---

[20] OMB, *Proposed Risk Assessment,* 16–17.

not satisfactory because partial disclosure precludes meaningful review by most reviewers, such as stakeholders, industry and community groups.

"In situations where public access to data and methods will not occur due to other compelling interests, agencies shall apply especially rigorous robustness checks to analytic results and document what checks were undertaken. Agency guidelines shall, however, in all cases, require a disclosure of the specific data sources that have been used."[21]

The bottom line is that full transparency and reproducibility of analysis is incompatible with the use of proprietary models. One solution would be for agencies to assert a prohibition on the use of proprietary models as the basis for public policy. In areas where no open-source models are currently available there are two plausible alternatives: either model proprietors would choose to release the source for their existing models, or other modelers would be commissioned to create new open-source alternatives.

IX. COMMUNITIES OF MODEL USERS AND DEVELOPERS

Open-source content collaborations, most notably Wikipedia, demonstrate that there are knowledgeable people willing to contribute remarkable efforts and creativity, unpaid and often with little public recognition, for the chance to participate in a collaborative project with real benefits to the world. One can easily imagine that substantial communities of contributors to policy modeling will grow around particular policy issues, such as health risks from air pollution, the mitigation of global warming or tax simplification.

Prior to the success of open-source projects, it was an industry truism that more programmers do not necessarily lead to faster development or better quality software.[22] Even programmers found it surprising that open-source projects could be so successful with so many contributors and little formal structure or project management.[23] It is also amazing, especially to publishers of proprietary encyclopedias, that Wikipedia boasts a high accuracy rate — its error rates are only slightly higher than *Encyclopædia Britannica* which

---

[21] OMB, "Guidelines for Ensuring," 8457.

[22] Frederick P. Brooks, *The Mythical Man-Month: Essays on Software Engineering* (New York: Addison-Wesley, 1995).

[23] Eric S. Raymond, "The Cathedral and the Bazaar."

offers vastly greater coverage, according to a recent study, given that anyone can contribute to and edit the text.[24]

It turns out that the internal organization of open-source projects is not as unstructured as outsiders may think. The structure is often informal and *ad hoc*. The best-known open-source software projects, Linux, Apache, MySQL, and Firefox, have millions of users. Anyone can download the source code, but relatively few actually do. Perhaps hundreds are actively involved in finding and fixing bugs. Typically, there are dozens of programmers who make significant contributions of new features. The core team that decides which changes do or do not go into the next release is even smaller. Often there is one person — most famously Linus Torvalds for Linux — who acts as a benevolent dictator who recruits and orchestrates the core team.

Wikipedia has hundreds of thousands of contributors and a much smaller community of editors who focus on reviewing changes in selected articles, especially the most controversial ones. While anyone can add or change text, the wiki technology makes it easy to back out edits that others do not deem valuable. The advantage of having a huge number of contributors is that there are many people (eyeballs) reviewing material who are likely to spot and remove undesirable additions quickly. This makes vandalism less appealing than physical graffiti, which may not be removed for months. Again, one benevolent dictator, Jimmy Wales, the originator of Wikipedia, helps resolve disputes and develop general rules for participants and editors.

Policy models, in contrast, typically have few "end users" who actually run the model. However, they can affect millions of people and billions of dollars in costs and benefits. Hence, there are often many people and organizations with strong interests in a model who are concerned about its assumptions, reliability, objectivity, and conclusions. Such organizations include government agencies, universities, industry groups, environmental and social justice organizations, and community groups, as well as individual citizens.

A limited number of people have the expertise to review and critique a policy model at the source code level; still fewer have the skills needed to build one. Some stakeholder organizations have members with the requisite expertise; others can hire them. Still others may find experts who are willing to contribute their skills pro bono. As the open-source software projects have demonstrated, it does not take a huge number of participants for a project to be successful, but it is important to develop a community of people who

---

[24] "Battle of Britannica: War has broken out between Encyclopædia Britannica and Nature," *The Economist* (March, 30, 2006): 65–66.

know each other by reputation, by contribution, or through email, wikis, and other web communications.

X. PATCHES AND PEDIGREES

Given the high stakes and the political and scientific controversies involved, will open-source policy models lead to clearer and more comprehensive models? Will they degenerate into incoherence, with incompatible model elements or be sabotaged by extremists?

If public agencies are to base policy decisions on model results, they need to be sure that the model is reliable and that changes are traceable. It might seem that an open-source model would preclude traceability, even existing open-source software licenses can include a mechanism to track who has made what contributions (sometimes known as the "pedigree"), such mechanisms allow users to select specific versions of the model that include or exclude particular contributions.

A major incentive for contributors to open-source projects, including Wikipedia, is that their contributions are public and visible to their peers. Every textual change or addition is linked to its author. The wiki technology makes it easy to compare versions. Even though open-source licenses allow anyone to modify and redistribute the code, they contain a provision that enables careful tracking of the editor's actions. According to The Open Source Definition which is the basis of many open source licenses:

> Encouraging lots of improvement is a good thing, but users have a right to know who is responsible for the software they are using. Authors and maintainers have reciprocal right[s] to know what they're being asked to support and [to] protect their reputations.[25]

This is the rationale for section four of the Open Source Definition, "Integrity of the Author's Source Code," which provides that, when an open-source license specifically precludes direct modification of the source code, the software creator must allow the distribution of the modification as a separate "patch" file.[26] This patch mechanism

---

[25] Ken Coar, "The Open Source Definition (Annotated)," *Open Source Initiative*, July 24, 2006, http://www.opensource.org/docs/definition.php (accessed November 7, 2007).

[26] Ibid., sub 4. A patch file changes or extends the functionality of the original code.

maintains the distinction between the original code and the modifications while attributing each element of the program to the proper author.

When there are many contributors and multiple versions tracking and managing them can be quite challenging. Open-source software projects generally use version management software, such as Subversion,[27] to track changes to modules, and create "releases" with an internally consistent set of module versions. Such tools would be essential for policy models in which responsibility for and control of version trees is critical. It may be necessary to extend such software with what we might call a "pedigree manager" that provides modelers and reviewers the ability to track each version and to change and manage the review, critique, and approval processes.

XI. CONTROVERSY AND DUELING MODELS

One major difference between policy modeling and other open-source software or Wikipedia is the intensity of controversy. In the case of policy models, the controversy often extends to disagreements about the models and science that inform them. In contrast, the functional goals of conventional software and Wikipedia articles are usually reasonably clear. Occasionally, when there are deep differences of opinion about goals and methods, open-source software projects have been known to fork into two or more versions with their attendant communities. Ultimately, the community of developers and the market of users determine which version succeeds and which falls by the wayside. Wikipedia, like conventional encyclopedias, is constitutionally focused on "accepted" knowledge. It requires a neutral point of view: articles should not advocate a point of view which is not generally accepted, although, on important areas of controversy, they may describe differing points of view as long as they do not advocate one view over another. Jimmy Wales calls this principle "absolute and non-negotiable."[28]

For policy models, user and developer communities with opposing views on a policy are liable to develop divergent models reflecting their different views. Forking may be the rule rather than the exception. It remains to be seen how this plays out as communities

---

[27] Collabnet Enterprise Edition, "Subversion," http://subversion.tigris.org (accessed November 7, 2007).

[28] Wikipedia, "Wikipedia: Neutral Point of View," *Wikipedia*, June 16, 2007, http://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view (accessed November 7, 2007).

develop around open-source policy modeling. It may prove hard to develop a single policy model relating to a controversial issue; two or more communities may emerge to create dueling models. Such counter-modeling can sometimes be informative and productive.

With policy models, unlike conventional software, it is often possible to combine dueling models as different versions or scenarios within a single meta-model that encompasses a broader set of possible assumptions. This approach makes it easier to compare and debate combinations of assumptions and their implications. Software that is capable of representing multiple alternatives or scenarios, along with pedigree management to identify who is responsible for the various elements, can greatly facilitate open-source policy modeling.

XII. STAKEHOLDER PERSPECTIVES

A commitment to open-source policy modeling would offer benefits and challenges for each kind of participant involved in policy making.

Government agencies and regulatory bodies that commission policy models and conduct policy analyses will receive several clear benefits. Open-source policy models enable a far more complete review by peers and stakeholders. The ability of reviewers to perform sensitivity analysis means that comments in response to rulemaking could be constructively focused on substantive issues, with fewer "trivial" criticisms. Model bugs can be found and fixed faster. Open source modeling ends the danger of agencies becoming "captive" to consultants with proprietary models. Greater model reusability and extensibility could make model development faster and cheaper. Making analyses more transparent can, of course, also make life more complicated for government policy makers and staff, as they have to deal with a wider range of more informed comments and critiques. In the US, however, the OMB has already committed itself to transparent regulatory analysis.

For stakeholder groups, including industry associations, community groups, and environmental and social justice groups, open-source modeling removes a crucial limit to the depth of their review. These groups can perform their own sensitivity analysis to critical assumptions. They can even extend a model to address omitted issues that they regard as important. They can become more active participants in the modeling process. Some industry groups and a few non-governmental organizations already have staff with the modeling expertise to do this, but many do not. The development of a wider community of modelers with expertise in particular models, or model

types, will make such experts more easily available as consultants to such organizations, paid or *pro bono*.

At universities and nonprofit think-tanks, some policy modelers already publish their models as open-source. As a source of modeling expertise and new techniques, as well as trainers of modelers, these organizations have much to contribute. If the models used by public agencies to support policy making and regulation are also open-source, it makes it easier for universities and think-tanks to be more intimately involved in reviewing and contributing to these models. This expands the community of expertise, improves the quality of the models, and could increase the chance that their contributions will have real influence on policy decisions.

Consulting firms with proprietary models may find this proposal disturbing. Some firms may be tempted to fight or delay its adoption. I suspect that others will find greater success by embracing it and demonstrating their ability to create transparent and extensible models for their clients. Governments could encourage firms to release existing proprietary models under open-source licenses by indemnifying them for any errors in previous analyses that may become apparent after release.

Open-source policy modeling will not eliminate the need for consultants to create, extend, and apply these models. Open-source software has not eliminated the need for consultants, but has in fact created major opportunities for software developers with expertise in open-source products to extend and adapt them for the needs of industry and government. Programmers who have made major contributions to creating high-quality open-source software often find lucrative contracts and jobs, as well as the satisfaction of recognition among their peers. The same could happen for effective open-source policy modelers.

As a citizen, it seems that open-source policy modeling offers compelling advantages for expanding the concept of transparency, a concept that is basic to the democratic process. It enables policy making to benefit from a wider community of reviewers, stakeholders, and contributors to models. It reduces the opportunity for policy making to be captured by a particular interest group to the exclusion of others. It offers the possibility of increasing the quality of policy models while reducing the expense of developing them.

XIII. CONCLUSIONS

Over time, open-source policy modeling has the potential to transform the way policy models are developed, the way they

influence public policy, and the way we arrive at the important decisions that we need to make as a society. The lessons from the open-source software movement and open collaborative projects such as Wikipedia suggest the ripple effects could be profound. Arguably, existing laws and policies in the US, including FOIA and OMB directives, already imply something close to open-source policy models, subject to the concerns of proprietary software, confidentiality, and security. However, these implications have not been clearly articulated, much less implemented. A more explicit commitment to the principle of open-source policy modeling would help. In the meantime, there is nothing to stop interested public agencies and stakeholder groups from exploring and demonstrating the benefits and challenges of the approach.